IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 17, 2021, accepted March 29, 2021, date of publication April 30, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3076849

Semi-Supervised Temporal Segmentation of Manufacturing Work Video by Automatically Building a Hierarchical Tree of Category Labels

KAZUAKI NAKAMURA[1], (Member, IEEE), NAOKO NITTA[1], (Member, IEEE), NOBORU BABAGUCHI[1], (Senior Member, IEEE), KENSUKE FUJII[2], SATOKI MATSUMURA[2], AND EIJI NABATA[2]

[1]Graduate School of Engineering, Osaka University, Osaka 565-0871, Japan
[2]Manufacturing Engineering Development Center, Komatsu Ltd., Osaka 573-1011, Japan

Corresponding author: Kazuaki Nakamura (k-nakamura@comm.eng.osaka-u.ac.jp)

This work was supported in part by the Graduate School of Engineering, Osaka University, and in part by the Manufacturing Engineering Development Center, Komatsu Ltd.

ABSTRACT Nowadays, many industrial companies visually record workers' activities for the purposes of streamlining their work processes. However, since untrimmed raw videos are hard to use, it is desired to automatically divide the videos into segments and recognize which kind of operation is performed on each segment. This task is called temporal video segmentation. We propose a method for achieving it, particularly targeting videos of manufacturing work with a specialized vehicle such as a hydraulic excavator. To make the performance of temporal video segmentation high, it is quite essential to extract good visual features from input videos. This can be hardly achieved by unsupervised methods, whereas supervised methods have another drawback that collecting a sufficient amount of training data is difficult due to its labor-intensiveness. To overcome these drawbacks, the proposed method employs a semi-supervised approach. We assume that a set of weakly-labeled videos whose frames only sparsely have a category label are given as input, where the labeled frames are used as training data to train a desirable feature extractor. Under this assumption, the proposed method first divides the input videos into short segments called primitive segments having the fixed length and then clusters them using visual features extracted by the above feature extractor. To achieve higher performance, we also use a hierarchical tree of the category labels and recursively perform the above process at each branch in the tree, where the tree is automatically built by the proposed method. In our experiments, we achieved a segmentation performance of 0.947 on the F-measure, even when only 1.25% of all the frames in the input videos are labeled.

INDEX TERMS Temporal video segmentation, semi-supervised clustering, hierarchical tree of category labels, manufacturing work video, deep metric learning.

I. INTRODUCTION

In today's society, large scale manufacturing work using a specialized vehicle such as a bulldozer, road roller, aerial platform, and so on is daily performed by many industrial companies. Typical examples of the manufacturing work include civil construction using a hydraulic excavator. Industrial companies conducting such work often have a strong demand to record workers' activity history in order to streamline their work process. For instance, in the case of civil construction with a hydraulic excavator, workers perform various operations such as *digging*, *swinging*, *dumping*, *idling*, and so on. Recording when and how long the workers perform each operation allows their manager to grasp their current situation, whose results are helpful for rearranging the process flow to make it more efficient. However, manually recording the activity history is quite labor-intensive, thus a technique for automatically recording it with inexpensive devices, especially a single camera, is desired and expected. In this paper, we focus on the vision-based recording of large scale manufacturing work with a specialized vehicle, assuming that a camera is installed in the vehicle and shoots its outside to

The associate editor coordinating the review of this manuscript and approving it for publication was Naveed Akhtar.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

FIGURE 1. Example of manufacturing work video.

observe work objects and scene from the first-person viewpoint, as shown in FIGURE 1. We refer to the videos captured by the in-vehicle camera as *manufacturing work videos*.

Vision-based activity recording can be realized by a technique of *temporal video segmentation*, which consists of the following two sub-tasks: *temporal division* and *categorization*. The former, temporal division, is a task of temporally dividing an input raw video into segments so that only a single kind of operation is performed in each segment. This is the first step of temporal video segmentation. Then, as the second step, the category label of the operation performed in each segment is recognized, which is the categorization.

There are many previous studies on temporal video segmentation that focused on human's daily activities such as sports actions and cooking [1]–[6]. These studies generally assume that a limited working area (e.g. cooking table) is shot by a stationary camera, where the background scene is fixed and detection techniques of the human body, hands, and objects play a basic role. Unlike them, we particularly focus on manufacturing work videos and propose a method for temporally segmenting them, where we cannot be based on a fixed background scene and detection-related techniques. To the best of our knowledge, there are only a few existing studies focusing on the analysis of manufacturing work videos.

Methods of temporal video segmentation are roughly divided into supervised ones [7], [8] and unsupervised ones [9], [10]. The former is advantageous in terms of performance; it allows us to obtain excellent segmentation results if a sufficient amount of training data is available, where the training data means fully-segmented video having frame-wise category labels. However, collecting a large set of fully-segmented videos is quite labor-intensive, especially in the case of manufacturing work, whose video is difficult for non-experts to manually annotate. Due to the disadvantage, supervised methods are not suitable for the analysis of manufacturing work videos in practical situations. On the other hand, unsupervised methods need no training data, which is their most important advantage, while their performance is relatively poor. This is because, not only hand-crafted features such as histogram of oriented gradients (HoG) [11] but also recent neural network-based features such as the one extracted by autoencoders [12] do not work well without training except for the case that quite simple human activities are targeted (e.g., just walking, running, jumping, and so on).

From the above reasons, we focus on a semi-supervised method that only needs weakly-labeled videos, where the term "weakly-labeled" means that the video itself is not pre-segmented while only a few frames in it are sparsely labeled. More specifically, given a set of weakly-labeled videos as input, our proposed method temporally divides them into segments and provides a category label for each segment. FIGURE 2 shows the relationship between the input and the output of the proposed method. The weakly-labeled videos can be collected with much less labor effort than fully-segmented videos. In addition, the weak-labels allow us to train a feature extractor that is helpful for temporal segmentation of manufacturing work videos, which is expected to drastically improve the performance. Hence, the proposed method overcomes the disadvantage of supervised methods and that of unsupervised ones.

Importantly, training of the feature extractor in the above process becomes more difficult when more diverse category labels are considered. This is because, a visual feature helpful for discriminating between two categories A and B is not necessarily consistent with that for discriminating between other two categories C and D. To solve this problem and further improve the performance of temporal video segmentation, we propose to use a hierarchical tree of category labels and recursively perform the above process at each branch in the tree, where the tree itself is also automatically built by the proposed method. This is our main contribution.

The other contributions of this work are summarized as follows.

- Manufacturing work with a specialized vehicle is targeted, in which any a priori knowledge such as human skin color, body/hand structure, fixed background scene, and so on is not available.
- A novel direction is provided to the task of temporal video segmentation: using weakly-labeled videos, i.e., videos consisting of sparsely labeled frames, as input.

The remainder of this paper is organized as follows. First, we briefly review previous studies related to this paper in Section II. Next, in Section III, we describe the proposed method of semi-supervised temporal video segmentation in detail, which is extended by introducing a hierarchical tree of category labels in Section IV. Then, the performance of the proposed method is experimentally evaluated in Section V, and finally, we conclude this paper in Section VI.

II. RELATED WORK

To the best of our knowledge, there are only a few studies focusing on temporal segmentation of manufacturing work video. Rather, methods for similar tasks have been widely studied in the field of human activity analysis, which we discuss in this section.

A. SUPERVISED ACTIVITY VIDEO ANALYSIS

Vision-based human activity analysis is one of the hottest topics in the field of computer vision and has been studied

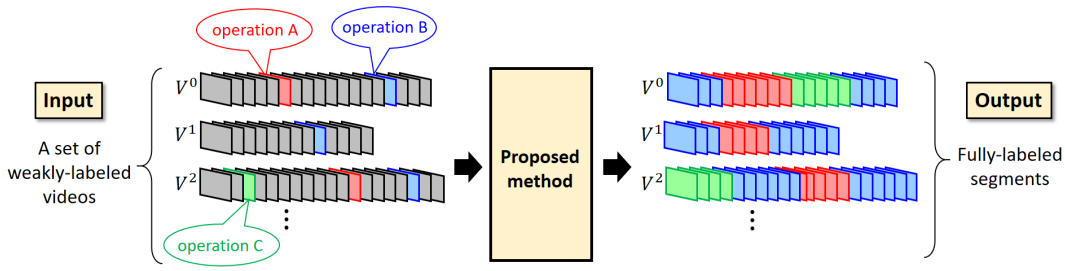

FIGURE 2. Task of semi-supervised temporal video segmentation targeted in this paper.

for more than 20 years. In early studies, pre-segmented videos consisting of fully-labeled frames were generally assumed as training data, and a recognition model based on hand-crafted features was trained, which was then used to recognize the category labels of unlabeled yet pre-segmented test videos. This is so-called *action recognition* task. Since the effectiveness of the hand-crafted features directly influences the overall recognition performance, a lot of well-designed features were proposed, whose instances include HoG [11], histogram of optical flows (HoF) [13], 3D Harris corners [14], 3D scale invariant feature transform (3D SIFT) [15], and so on. These features are generally extracted from spatio-temporal local regions in an input video and integrated into a video-wise feature vector with *bags of visual word* (BoVW) strategy [16].

Of course, the above methods were not practical because pre-segmented videos are difficult to obtain in realistic situations. Hence, researchers' interest shifted from action recognition to *action segmentation*, which assumes unsegmented videos as input and simultaneously solves temporal division and categorization sub-tasks. Wang *et al.* proposed a sliding window-based method, which slides a template segment of the targeted human action over an input video to find its sub-sequences matched to the template [7]. Hara *et al.* proposed a voting-based method, which detects temporal boundaries between sequentially performed actions by modeling the relationship between local features and the temporal action boundaries [8].

More recently, convolutional neural networks (CNNs) were introduced into the task of activity analysis. The focus of these methods includes not only temporal segmentation but also spatial segmentation; that is, they attempt to specify the spatio-temporal region in a video where each of the targeted human actions is performed. This novel task is called *action localization*. Hou *et al.* proposed a 3D-CNN-based action localization method called T-CNN [17], which is an extension of R-CNN, a method for detecting objects from a still image with 2D-CNN. Since most human actions are characterized by their appearance and motion, Zhao *et al.* proposed to use two-streams network architecture for action localization, one of which directly handles each frame in an input video and the other handles optical flow field computed between two neighboring frames [18]. Moreover, to represent the temporal structure of human actions more effectively, Ramaswamy *et al.* focused on *long short term memory* (LSTM) and proposed to use a hierarchical LSTM for action localization [19].

As seen above, supervised methods of human activity analysis have been widely studied. Although this field is still growing, the supervised methods are not suitable for practical applications due to their labor-intensiveness for collecting training data. This is a serious problem in the case of manufacturing work video because it is difficult for non-experts of the work to manually provide the correct ground truth label to every frame in training videos.

B. UNSUPERVISED ACTIVITY VIDEO ANALYSIS

Early studies on unsupervised activity analysis were focusing on pre-segmented videos, similar to those on supervised methods. Since no label information is available, unsupervised methods generally employ a clustering approach; they first extract a feature vector from each segment in an input video set, then cluster the extracted features. Each of the resultant clusters corresponds to a single action category, whose name is manually given after the clustering process. As the segment-wise feature, Dollar *et al.* proposed to use BoVW representation of spatio-temporal local descriptors [20]. Niebles *et al.* also focused on spatio-temporal local descriptors, and employed two famous latent topic models called *probabilistic latent semantic analysis* and *latent Dirichlet allocation* for the clustering process [21].

On the basis of the above studies, methods focusing on unsegmented videos, i.e., unsupervised action segmentation methods, were next studied. The purpose of this task is to temporally divide input videos into segments and cluster them. Once a result of the temporal division is given, the subsequent clustering process can be straightforwardly done. In addition, clustering techniques generally provide not only a clustering result but also its score, which evaluates how much the resultant clusters are compact as well as differ from each other. Based on these facts, Zhou *et al.* formulated the task of unsupervised action segmentation as a maximization problem of the clustering score [9]; their method divides input videos into segments so that the score of the clustering results for them is maximized. Kruger *et al.* employed a similar approach and extended it [10].

Recently, unsupervised methods for action localization have become a hot topic, whose first work is made by Soomro's team [22]. Their method extracts spatio-temporal local features from input videos and clusters them to obtain representative action categories and their instances. Then a discrimination model for the action categories is trained by

a support vector machine, which is used for refining the instance sets of the action categories. This process is iteratively repeated. Gong *et al.* employed a similar approach, using a CNN with a temporal attention mechanism as the discrimination model [23]. Unlike these methods, Jain *et al.* proposed to extract spatio-temporal action regions called *tubelets* from input videos at once [24]. The tubelets are extracted by dividing input videos into super-voxels and grouping them.

To make unsupervised methods including the above ones work well, we have to design a visual feature that is suitable to the target activity. For human activities, several ways to obtain such a good feature have been explored through a massive amount of previous work. For instance, CNNs trained on various benchmark datasets of human action videos could be helpful. However, for the activities not widely studied yet, there is no benchmark dataset and it is quite difficult to design a good visual feature. Hence, a method for temporal video segmentation that can achieve high performance is hardly obtained in an unsupervised manner.

C. SEMI-SUPERVISED ACTIVITY VIDEO ANALYSIS

There are several previous studies focusing on semi-supervised or weakly-supervised action localization [25]–[27]. However, between these studies and ours, the meaning of the term "semi-supervised" is different. The general scenario of semi-supervised action localization is as follows. Input data is a set of videos, each of which consists of action-related segments and other background segments. The boundaries between the two kinds of segments are unknown; the input videos are not pre-segmented. Note that only a single kind of action is performed in each input video, whose category label is given. This is called *video-level supervision*. The purpose of the action localization methods is to selectively spot the action-related segments from the input videos. Obviously, this scenario is different from ours shown in FIGURE 2, in which two or more kinds of operations are performed in each video. Hence, the existing semi-supervised methods cannot be directly applied to our target task.

III. SEMI-SUPERVISED TEMPORAL SEGMENTATION OF MANUFACTURING WORK VIDEO

In this section, we describe the basic approach employed in the proposed method, which regards the task shown in FIGURE 2 as the problem of semi-supervised clustering. The approach consists of several steps. We start from the explanation of its overview in Subsection III-A and then proceed to the details of each step in the subsequent subsections.

A. OVERVIEW

As shown in FIGURE 2, a set of weakly-labeled videos is given as input in our scenario. Let $\mathcal{V} = \{V^0, V^1, \cdots\}$ be the set, where V^i is the i-th input video in which several operations are sequentially performed. Each video is formulated as a sequence of frames, i.e., $V^i = (f_0^i, f_1^i, \cdots, f_{n-1}^i)$, where f_j^i is the j-th frame in the V^i and n is its length. Note that only a few frames in $\mathcal{F} = \{f_j^i \mid i = 0, 1, \cdots, j = 0, 1, \cdots\}$ have a category label, or an operation name, given as semi-supervision. We refer to the label of frame $f \in \mathcal{F}$ as $L(f)$ if it is given.

To achieve temporal video segmentation for the $\mathcal{V}$, we employ the following approach consisting of three steps.

(1) Temporally divide each video V^i into primitive segments $p_0^i, p_1^i, p_2^i, \cdots$, where all the primitive segments have the fixed length t and do not overlap with each other; the k-th primitive segment p_k^i is just the sequence of frames represented as $p_k^i = (f_{kt}^i, f_{kt+1}^i, \cdots, f_{kt+t-1}^i)$.
(2) Apply a clustering algorithm to a set of the primitive segments $\mathcal{P} = \{p_k^i \mid i = 0, 1, \cdots, k = 0, 1, \cdots\}$ so that different clusters correspond to different categories.
(3) Concatenate successive primitive segments into a single segment if they are assigned to the same cluster.

In step (2), we can make each primitive segment have at most one labeled-frame by setting the length t sufficiently short. If the primitive segment p_k^i has a labeled frame f, we consider the p_k^i as labeled data, whose category label is defined as $L(p_k^i) = L(f)$. Otherwise, we consider the p_k^i as unlabeled data. This consideration allows us to straightforwardly regard step (2) as the problem of semi-supervised clustering. FIGURE 3 shows the overall picture of the approach.

Hereafter, we focus only on step (2), where we can drop the superscript i from the above notation without loss of generality. Thus, we rewrite a set of the primitive segments as $\mathcal{P} = \{p_k \mid k = 0, 1, \cdots, N-1\}$, where N is the total number of the primitive segments. In addition, we assume that only the first M ($\ll N$) primitive segments are labeled. Under this assumption, we achieve step (2) as follows.

(2-a) Train a sophisticated feature extractor E that extracts a feature vector $\boldsymbol{x} = E(p)$ from a primitive segment p, by using a set of the labeled data $\mathcal{Q} = \{p_k \mid k = 0, 1, \cdots, M-1\} \subset \mathcal{P}$ as a training dataset.
(2-b) Extract a feature vector $\boldsymbol{x}_k = E(p_k)$ from p_k for all $k \in \{0, 1, \cdots, N-1\}$, and apply a clustering algorithm to a set of the extracted vectors $\mathcal{X} = \{\boldsymbol{x}_k \mid k = 0, 1, \cdots, N-1\}$.

We describe the details of step (2-a) and step (2-b) in Subsections III-B and III-C, respectively.

B. TRAINING OF FEATURE EXTRACTOR VIA DEEP METRIC LEARNING

In semi-supervised clustering, two labeled data should be assigned to the same cluster if they have the same category label, otherwise they should be assigned to different clusters. Hence, it is desirable to extract similar features from two data when and only when they share the same label. This can be realized by deep metric learning (DML) [28]. We employ a DML technique proposed by Hoffer *et al.*, which uses a deep neural network with a triplet loss function [29].

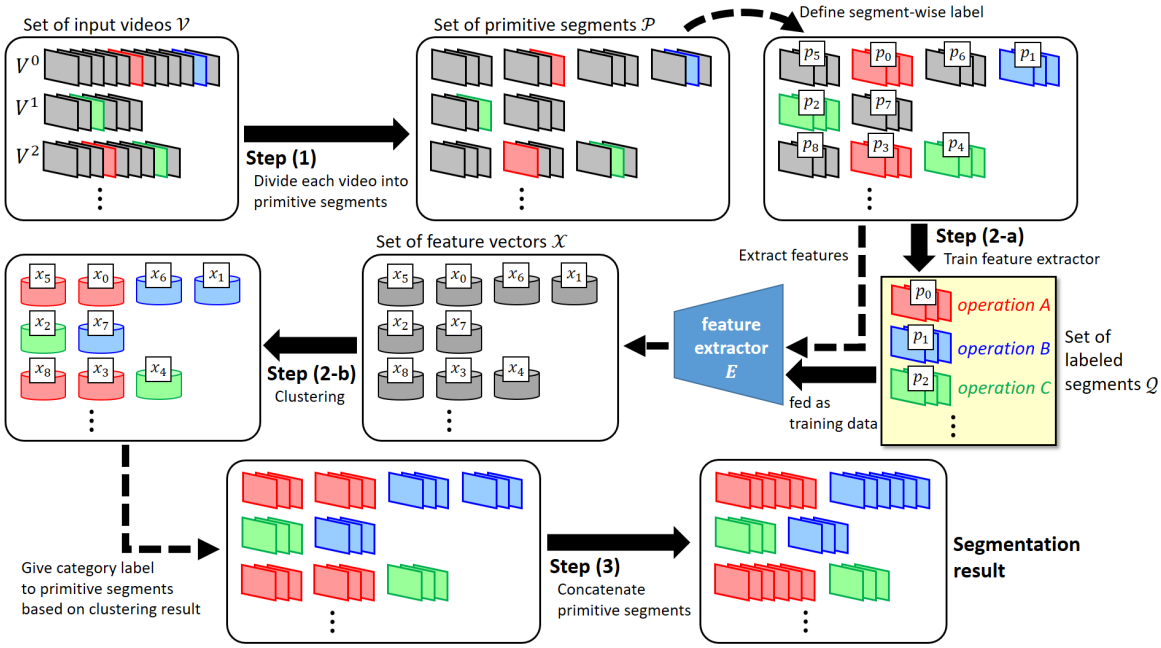

FIGURE 3. Overview of our basic approach consisting of three steps.

The aim of our DML process is to train a CNN-based feature extractor E that can extract a fixed-dimensional feature vector $\boldsymbol{x} = E(p)$ from a primitive segment p. To this end, we first generate a lot of triplets using a set of labeled segments $\mathcal{Q}$, where each triplet can be denoted by $(p^a,\ p^+,\ p^-)$. p^a is called *anchor* and randomly selected from $\mathcal{Q}$. p^+ and p^- are also randomly selected from $\mathcal{Q}$ so that p^+ has the same label with the anchor while p^- has a different label, namely $L(p^+) = L(p^a)$ and $L(p^-) \neq L(p^a)$. Next, based on the generated triplets, a loss function

$$\mathrm{Loss}(E) = \sum_{(p^a,\ p^+,\ p^-)} \frac{d(p^a, p^+; E)}{d(p^a, p^+; E) + d(p^a, p^-; E)} \tag{1}$$

is defined, where

$$d(p, p'; E) = \exp ||E(p) - E(p')|| \tag{2}$$

is the dissimilarity score between two primitive segments p and p'. $\mathrm{Loss}(E)$ becomes smaller if $d(p^a, p^+; E)$ is smaller as well as $d(p^a, p^-; E)$ is larger. Hence, minimizing the loss function with respect to E, in other words, finding the optimal E^* that satisfies $E^* = \mathrm{argmin}_E \mathrm{Loss}(E)$, we can get a desirable feature extractor.

Since primitive segment p is a sequence of video frames and therefore is large-sized data, we reduce its dimensionality by compressing each frame f_j in the $p = (f_0,\ \cdots,\ f_{t-1})$ into a m-dimensional vector $\boldsymbol{y}_j$ and make a matrix $Y = (\boldsymbol{y}_0\ \cdots\ \boldsymbol{y}_{t-1})$ whose size is $m \times t$. The matrix Y is actually used as an input to the feature extractor E instead of the p. Note that the vector $\boldsymbol{y}_j$ should be obtained in an unsupervised manner. In the experiments described later in Section V, we obtain it by an autoencoder whose input/output is a combination of the frame f_j and the optical flow field calculated

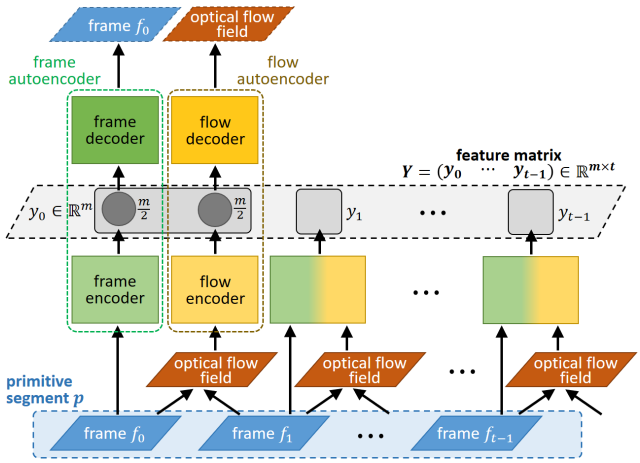

FIGURE 4. Dimensionality reduction from primitive segment p into matrix Y.

between f_j and f_{j+1}, as show in FIGURE 4. The network structure of the feature extractor E is shown in FIGURE 5.

C. CLUSTERING OF PRIMITIVE SEGMENTS

Since the feature vector $\boldsymbol{x}$ extracted by the above E is just a vector of real values, any clustering algorithm can be applied to $\mathcal{X} = \{\boldsymbol{x}_k \mid k = 0, 1, \cdots, N-1\}$ to cluster them. In the proposed method, we employ spectral clustering [30] for the purpose, whose procedure is briefly summarized as follows. First, we construct a complete undirected graph whose nodes are corresponding to the feature vectors one-to-one. The weight of the edge between two nodes corresponding to $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ is given as their similarity score $\mathrm{sim}(\boldsymbol{x}_k, \boldsymbol{x}_l) = \exp(-\beta||\boldsymbol{x}_k - \boldsymbol{x}_l||^2)$, where β is a positive constant. Then, we divide a set of the nodes into several subsets so that

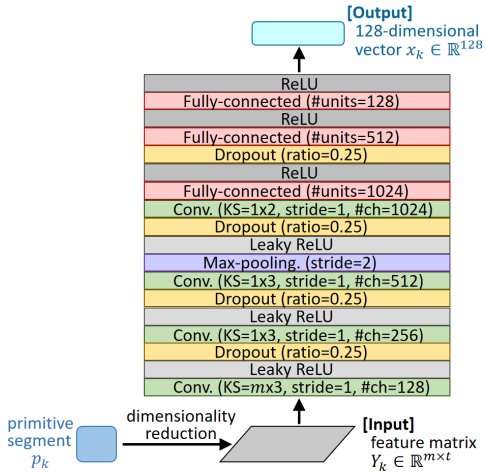

FIGURE 5. Network structure of feature extractor E. "KS" and "#ch" means kernel size and number of channels of convolution filter, respectively. "#units" means number of units in fully-connected layer.

nodes in the same subset have heavily weighted edges while nodes in different subsets have lightly weighted edges. Each subset in the division result corresponds to a single cluster. See the original paper [30] for the details of the division process.

IV. EXTENSION BY BUILDING A HIERARCHICAL TREE OF CATEGORY LABELS

Practically, a feature extractor provided by DML does not always work well. Suppose the case that there are four different classes having category labels A, B, C, and D, and two-dimensional datapoints belonging to each class are normally distributed with an identity covariance matrix, as shown on the left side of FIGURE 6. The mean of the datapoint distribution of class A, B, C, and D is (0, 0), (0, 1), (1, 0), and (1, 1), respectively. The aim of DML is to design a one-dimensional feature space that is useful for discriminating between the four classes, which is almost impossible in this case. This is because the feature space best for discriminating between A and B is totally different from that for discriminating between A and C. This is also the case with any other combination of three classes. This problem becomes more serious when the number of classes increases more.

A promising solution for the above problem is to introduce a hierarchical tree of category labels, as shown on the right side of FIGURE 6. We abbreviate it as *label tree* in the remainder. If we first marge the classes A and B into A' and the classes C and D into C' based on the label tree, we can easily obtain a feature space useful for discriminating between A' and C' by DML. Then, by applying DML again to A' and C' separately, we can obtain another feature space useful for discriminating between A and B and that useful for discriminating between C and D. However, it is not easy to manually define a good label tree in a practical situation where more number of classes (or category labels) should be assumed. Hence, in this section, we propose a method for

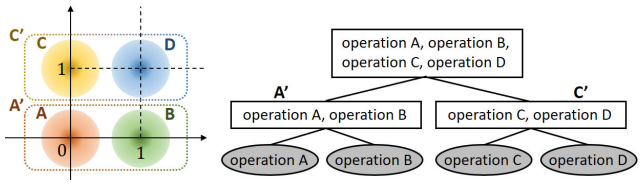

FIGURE 6. Example case where DML aims to find feature space helpful for discriminating between four classes A, B, C, and D. [Left] Datapoint distribution of each class, for which it is almost impossible to make single desirable feature space. [Right] Hierarchical tree of category labels, by which DML can separately obtain three different feature spaces that are helpful for discriminating between A' and C', A and B, and C and D.

automatically building a good label tree to extend the method proposed in the previous section.

A. OVERVIEW

Let $\mathcal{L}$ be a set of the category labels given as frame-level semi-supervision. For instance, $\mathcal{L} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ in the case of the situation in FIGURE 6. Using $\mathcal{L}$ and $\mathcal{P} = \{p_k \mid k = 0, \cdots, N-1\}$, we build a label tree by the following six-steps algorithm, which limits its structure to a binary tree.

(i) Randomly partition the label set $\mathcal{L}$ into two subsets $\mathcal{K}_0$ and $\mathcal{K}_1$ such that $\mathcal{K}_0 \cap \mathcal{K}_1 = \emptyset$ and $\mathcal{K}_0 \cup \mathcal{K}_1 = \mathcal{L}$. For instance, $\mathcal{K}_0 = \{\mathrm{A}, \mathrm{D}\}$ and $\mathcal{K}_1 = \{\mathrm{B}, \mathrm{C}\}$.

(ii) Give a pseudo label s_k for each $p_k \in \mathcal{P}$ if it has a category label. More specifically, the pseudo label s_k is defined as

$$s_k = \begin{cases} 0 \ (\text{if} \ \ L(p_k) \in \mathcal{K}_0) \\ 1 \ (\text{if} \ \ L(p_k) \in \mathcal{K}_1) \end{cases}. \tag{3}$$

(iii) Apply the DML technique proposed in Subsection III-B to a set of the labeled primitive segments $\{(p_k, s_k) \mid k = 0, \cdots, M-1\}$ in order to train a feature extractor, which is next used for extracting a feature vector from p_k for all $k \in \{0, \cdots, N-1\}$. Then, using the extracted feature vectors, divide a set of the primitive segments $\mathcal{P}$ into two clusters by the technique described in Subsection III-C.

(iv) Evaluate the clustering result of step (iii) under a certain criterion. After the evaluation, discard the clustering result.

(v) Repeat steps (i)-(iv) many times to find the best partition of $\mathcal{L}$, i.e., $(\mathcal{K}_0, \mathcal{K}_1)$ that acquires the highest evaluation score in the step (iv). Let $(\hat{\mathcal{K}}_0, \hat{\mathcal{K}}_1)$ be the best partition. For instance, $\hat{\mathcal{K}}_0 = \{\mathrm{A}, \mathrm{B}\}$ and $\hat{\mathcal{K}}_1 = \{\mathrm{C}, \mathrm{D}\}$. Then, perform the steps (ii) and (iii) again under the best partition, by which the $\mathcal{P}$ is actually divided into two clusters $\mathcal{P}_0$ and $\mathcal{P}_1$.

(vi) Recursively apply this algorithm to $\hat{\mathcal{K}}_0$ and $\mathcal{P}_0$ as well as $\hat{\mathcal{K}}_1$ and $\mathcal{P}_1$, until all the category labels in $\mathcal{L}$ belong different subsets from each other.

This is a kind of hierarchical clustering algorithm, which allows us to not only build a label tree but also obtain a clustering result on $\mathcal{P}$.

Note that it is needed in practical situations to efficiently find the best partition $(\hat{\mathcal{K}}_0, \hat{\mathcal{K}}_1)$, whose way is described in

the next subsection. After that, we describe the evaluation criterion employed in step (iv) in Subsection IV-C.

B. FINDING BEST PARTITION OF CATEGORY LABEL SET VIA GENETIC ALGORITHM

The problem of finding the best partition $(\hat{\mathcal{K}}_0, \hat{\mathcal{K}}_1)$ can be regarded as a discrete optimization problem by the following formulation. Let r denote the number of unique labels in $\mathcal{L}$, i.e., $r = |\mathcal{L}|$. A partition $(\mathcal{K}_0, \mathcal{K}_1)$ can be represented by a r-dimensional binary vector $\boldsymbol{g} \in \{0, 1\}^r$, whose j-th dimension is defined as 0 if the j-th label in the $\mathcal{L}$ belongs to $\mathcal{K}_0$; otherwise it is defined as 1. Let $\mathrm{scr}(\boldsymbol{g})$ be the evaluation score for the clustering result under the partition represented by $\boldsymbol{g}$, which is computed in the step (iv) in the algorithm of the previous subsection. With this formulation, we can obtain the best partition $(\hat{\mathcal{K}}_0, \hat{\mathcal{K}}_1)$ by finding $\boldsymbol{g}^* = \mathrm{argmax}_{\boldsymbol{g}}\{\mathrm{scr}(\boldsymbol{g})\}$. To efficiently solve this optimization problem, we employ a genetic algorithm.

First, r-dimensional binary vectors $\boldsymbol{g}_0, \boldsymbol{g}_1, \cdots$ are randomly generated, which are used as the individuals in the initial generation. Next, for each individual $\boldsymbol{g}_i$ $(i = 0, 1, \cdots)$, its score $\mathrm{scr}(\boldsymbol{g}_i)$ is computed. This is no other than steps (ii)-(iv) in the algorithm of the previous subsection. Then, half of the individuals having lower scores are deleted, and the remaining half having higher scores survives to the next generation. Moreover, genetic operators named *crossover* and *mutation* are performed on the survived individuals, whose results are added into the next generation. We repeat the above procedure until the number of generations reaches a pre-determined threshold, and finally, select the individual that has the highest score in the last generation.

C. EVALUATION CRITERION FOR CLUSTERING RESULT

We need to calculate $\mathrm{scr}(\boldsymbol{g})$ to run the above genetic algorithm. For this purpose, we employ *modularity* [31], which is an evaluation criterion for a clustering result.

Let $E^{\boldsymbol{g}}$ be a feature extractor trained by DML under the condition of $\boldsymbol{g}$, and let $\boldsymbol{x}_k^{\boldsymbol{g}} = E^{\boldsymbol{g}}(p_k)$ be the feature vector extracted by $E^{\boldsymbol{g}}$ from the primitive segment p_k. Moreover, Let $c(\boldsymbol{x}_k^{\boldsymbol{g}})$ be the ID of the cluster that $\boldsymbol{x}_k^{\boldsymbol{g}}$ was assigned to. Based on a set of indexes

$$\mathcal{N}_k^{\boldsymbol{g}} = \{l \mid c(\boldsymbol{x}_l^{\boldsymbol{g}}) = c(\boldsymbol{x}_k^{\boldsymbol{g}})\} \setminus \{k\} \tag{4}$$

and its complement set

$$\overline{\mathcal{N}}_k^{\boldsymbol{g}} = \{l \mid c(\boldsymbol{x}_l^{\boldsymbol{g}}) \neq c(\boldsymbol{x}_k^{\boldsymbol{g}})\} \tag{5}$$

the modularity is calculated as

$$\mathrm{scr}(\boldsymbol{g}) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{1}{\mathcal{N}_k^{\boldsymbol{g}}}\sum_{l\in\mathcal{N}_k^{\boldsymbol{g}}}\mathrm{sim}(\boldsymbol{x}_k^{\boldsymbol{g}}, \boldsymbol{x}_l^{\boldsymbol{g}}) - \frac{1}{\overline{\mathcal{N}}_k^{\boldsymbol{g}}}\sum_{l\in\overline{\mathcal{N}}_k^{\boldsymbol{g}}}\mathrm{sim}(\boldsymbol{x}_k^{\boldsymbol{g}}, \boldsymbol{x}_l^{\boldsymbol{g}})\right], \tag{6}$$

where $\mathrm{sim}(,)$ is the metric for measuring similarity score between two feature vectors used in the spectral clustering in Subsection III-C.

V. EXPERIMENTS

To evaluate the performance of the proposed method, we conducted experiments on a video set of civil construction work with a hydraulic excavator, which is a typical example of manufacturing work videos. The results of the experiments are described in detail in this section.

A. EXPERIMENTAL SETUP

1) DATASET AND PARAMETER SETTING

We prepared 11 videos where a worker performs actual civil construction work by driving a hydraulic excavator. These videos were captured with a camera installed in the hydraulic excavator. It is different in each video when, where, and which worker drove the hydraulic excavator. The length of the videos ranged from three to eight minutes, whose frame rate was 30 fps.

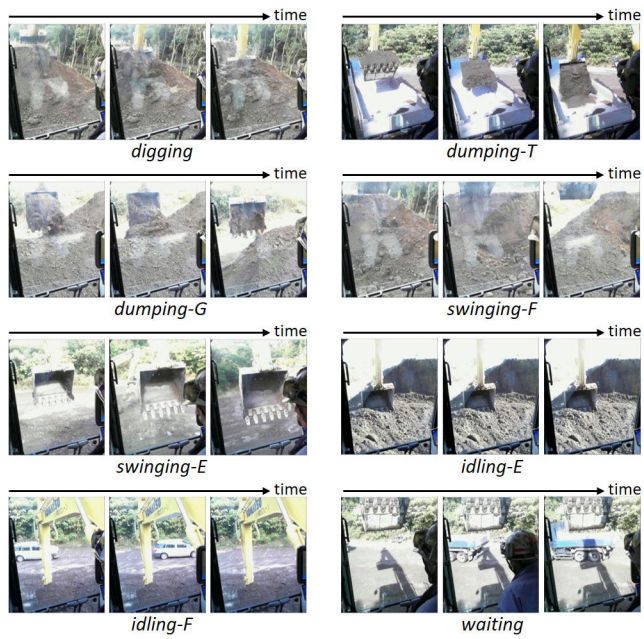

FIGURE 7. Example frames of eight kinds of civil construction operations.

We showed the videos to an expert worker of civil construction and asked him to define a set of category labels. As a result, the worker listed eight unique categories shown in TABLE 1 and gave one of these labels to each frame in the videos. Thus we collected the fully annotated videos having frame-wise labels. Next, we randomly selected η frames per category from the video set and used them as the labeled input for our target task. The labels given to the other frames were used just for evaluation. FIGURE 7 shows some example frames of the eight categories of civil construction operations.

In this experiment, we ran the proposed method on the above dataset under the settings of $\eta = 25, 50, 75, 100, 125,$

TABLE 1. Category labels (or operation names) listed by expert of civil construction work. "#frames" indicates number of frames having corresponding label in our dataset that consists of 11 videos.

ID	Name	Description	#frames
1	*digging*	digging dirt with the bucket	20,490
2	*dumping-T*	dumping dirt into a dump truck	17,490
3	*dumping-G*	dumping dirt onto the ground	3,720
4	*swinging-F*	swinging the arm with the bucket filled	15,810
5	*swinging-E*	swinging the arm with the bucket empty	16,650
6	*idling-F*	idling with the bucket filled	4,890
7	*idling-E*	idling with the bucket empty	6,450
8	*waiting*	waiting for a dump truck to arrive	10,950

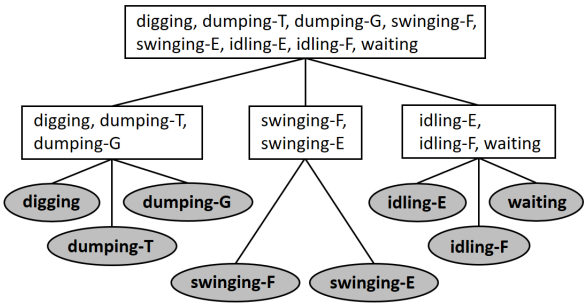

FIGURE 8. Hierarchical tree of category labels used in *SS+LTm*, which is defined by expert worker of civil construction.

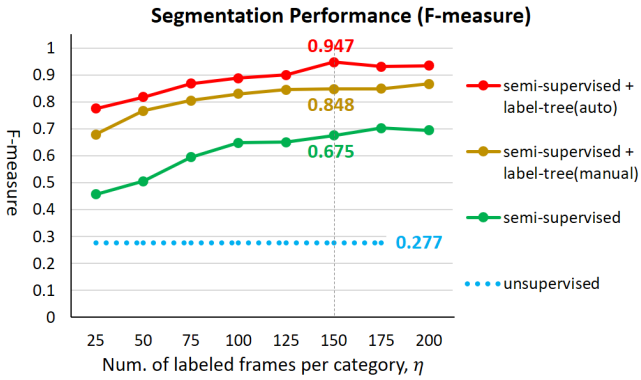

FIGURE 9. Performance comparison between *UN*, *SS*, *SS+LTm*, and *SS+LTa* under various settings of η from aspect of F-measure.

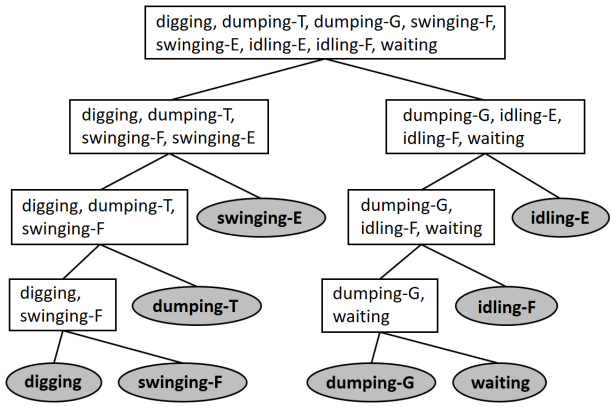

FIGURE 10. Hierarchical tree of category labels automatically built by proposed method, i.e., *SS+LTa*. This was obtained under setting of $\eta = 150$.

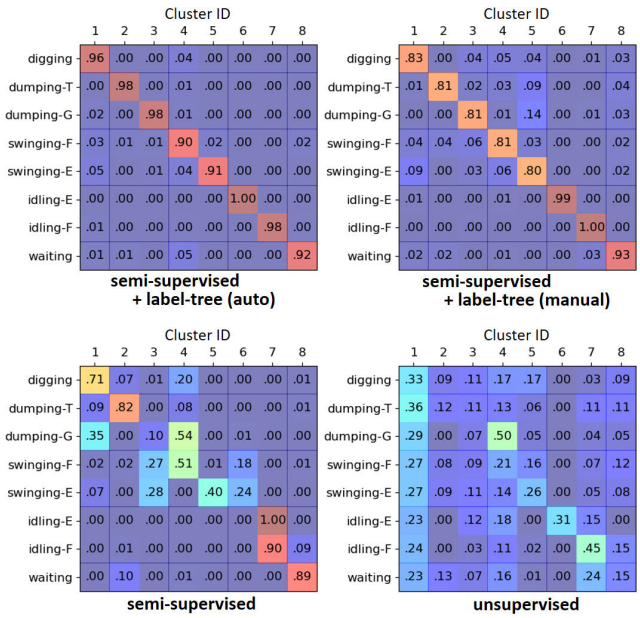

FIGURE 11. Confusion matrices of segmentation results obtained by *UN*, *SS*, *SS+LTm*, and *SS+LTa*, which are calculated by frame-wise ground truth labels and cluster IDs assigned by each method.

150, 175, and 200. The length of primitive segments, t, was set as 30. The number of generations and the population size per generation in the genetic algorithm were set as three and eight, respectively.

2) EVALUATION CRITERION

We regard the task of temporal video segmentation as the problem of semi-supervised clustering, as mentioned in Subsection III-A. Hence, to evaluate the segmentation result, we employed F-measure [32], which was a representative evaluation criterion for clustering tasks.

Let θ_{ij} be the number of frames that is assigned to the i-th cluster in step (2) of the proposed method and whose ground truth label is the j-th category. To compute the F-measure, we have to calculate *precision* and *recall* for each i and j, which are defined as

$$P_{ij} = \frac{\theta_{ij}}{\sum_{k=1}^{r} \theta_{ik}} \text{ and } R_{ij} = \frac{\theta_{ij}}{\sum_{k=1}^{r} \theta_{kj}} \tag{7}$$

respectively. Note that r is the number of unique ground truth labels ($r = |\mathcal{L}| = 8$ in our experiments). Next,

$$F_j = \max_i \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}} \tag{8}$$

is calculated for each j, and finally the F-measure is computed as

$$F = \sum_{j=1}^{r} \left[\frac{\sum_{k=1}^{r} \theta_{kj}}{\sum_{l=1}^{r} \sum_{k=1}^{r} \theta_{kl}} F_j \right]. \tag{9}$$

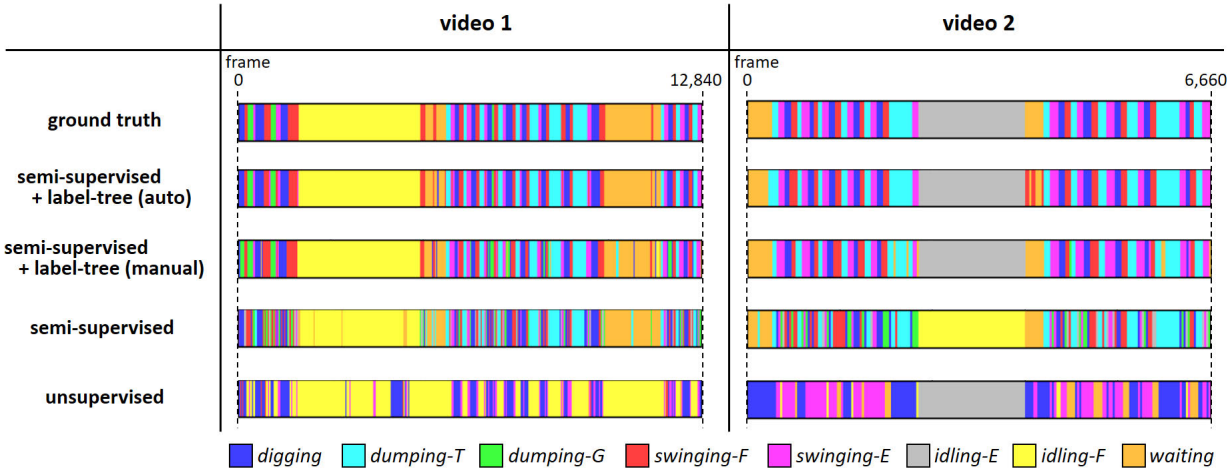

FIGURE 12. Examples of qualitative results of temporal video segmentation. This result was obtained when $\eta = 150$.

Values of the F-measure range from 0 to 1. A higher value of F-measure indicates a better segmentation result, thus $F = 1$ means the best.

3) COMPARED METHODS

For the purpose of comparison with the proposed method, we also ran the following three methods on our dataset.

- *Semi-supervised + label-tree (manual)*, abbreviated as *SS+LTm*, is almost the same method with the proposed one but the hierarchical tree of category labels is preliminary given, which is shown in FIGURE 8. This was manually defined by an expert worker of civil construction.
- *Semi-supervised*, abbreviated as *SS*, is a method not using any label tree, which divides a set of primitive segments into r clusters at once using a single feature extractor trained by DML. This is no other than the method described in Section III.
- *Unsupervised*, abbreviated as *UN*, is a method not employing DML. In this method, a matrix Y described in Subsection III-B is directly used as a feature of each primitive segment, and then the clustering approach described in Subsection III-C is applied to a set of the features.

In the remainder, we refer to the proposed method as *semi-supervised + label tree (auto)*, abbreviated as $SS + LTa$.

B. RESULTS AND DISCUSSION

FIGURE 9 shows the values of the F-measure for each method under various settings of η. Since *UN* does not use any labeled input, it was run only once, whose result is indicated by the blue dash line. As seen in this figure, *UN* is outperformed by *SS* even in the case of lower η, which is further outperformed by *SS+LTm* and *SS+LTa* under any setting of η. These results indicate the effectiveness of the semi-supervised learning as well as that of the label trees. Interestingly, *SS+LTa* always outperforms *SS+LTm*. This means that the automatically built label trees, whose example is shown in FIGURE 10, are more helpful for achieving temporal video segmentation than the expert-defined tree shown in FIGURE 8. The semantic similarity between two different categories is not always consistent with their visual similarity. Hence, a manually defined label tree is not necessarily the optimal one for temporal video segmentation. In contrast, the method proposed in Section IV tries to find the optimal tree based on visual features. This allows us to obtain a better segmentation result.

A value of the F-measure basically tends to be higher with larger η, but it is not so improved when $\eta \geq 150$. This indicates that 150 labeled frames per category are enough for obtaining a good segmentation result by the proposed method, which is quite a few compared with the total number of frames shown in TABLE 1. Only 1.25% of all the frames are labeled even in the case of $\eta = 150$, where the proposed *SS+LTa* achieved an F-measure of 0.947. *SS+LTm*, *SS*, and *UN* achieved a F-measure of 0.848, 0.675, and 0.277, respectively, under the same setting of η. This means that *SS+LTa* improves the performance of temporal video segmentation by 11.7%, 40.3%, and 242% compared with *SS+LTm*, *SS*, and *UN*, respectively.

FIGURE 11 shows the confusion matrices of the segmentation results obtained by *UN*, *SS*, *SS+LTm*, and *SS+LTa* when $\eta = 150$. The eight categories of civil construction operations cannot be successfully separated without label trees. *SS+LTm* improves the performance by introducing the manually defined label tree, however, *digging* and *dumping* operations are still sometimes confused with *swinging* operation. The scores of the three operations in the confusion matrix of *SS+LTm* are at most 0.83. This is because these operations are essentially difficult to separate from each other since they could be seamlessly performed. Surprisingly, this problem is partially solved in the result of *SS+LTa*, where the scores of *digging*, *dumping*, and *swinging* operations in the confusion matrix exceed 0.9. This result indicates the effectiveness of the automatically-built label trees.

We finally examined the qualitative results of temporal segmentation for the 11 videos, two of which are picked out and shown in FIGURE 12. The horizontal axis of each band in the figure indicates the time axis, or frame ID, and the colors of the band indicate the category labels. We can see in this figure that the proposed *SS+LTa* can divide the two videos into segments as almost same as the ground truth.

VI. CONCLUSION

In this paper, we proposed a method for temporal video segmentation, particularly targeting videos of large-scale manufacturing work with a specialized vehicle. To reduce the labor-intensiveness of training data collection as well as realize a good segmentation performance, we employ a semi-supervised approach. Given a set of weakly-labeled videos whose frames are only sparsely labeled, the proposed method trains a feature extractor using the labeled frames as training data. At the same time, it divides the input videos into primitive segments having the fixed length and clusters them using the visual features extracted by the feature extractor. To further improve the performance, we iteratively perform the above process according to a hierarchical tree of category labels, where the tree is automatically built by the proposed method.

We examined the effectiveness of the proposed method through experiments, whose findings are summarized as follows.

- Label trees play an important role in accelerating the performance of temporal video segmentation; the proposed method improved the segmentation performance by 40.3% compared with its baseline approach not using any label tree.
- Label trees automatically built by the proposed method are more helpful than those defined by experts; a performance improvement of 11.7% was achieved by the automatically built trees.
- Only a small ratio of labeled data is enough for the proposed method to work well; it achieved an F-measure of 0.947 even when 1.25% of all the frames in the input video are labeled.

The proposed method has a limitation that it cannot handle unknown category labels. For instance, when four kinds of operations A, B, C, and D are actually performed in the input videos but there are no frames labeled as D, our current method does not work well. This limitation could be overcome by interaction with a human annotator, in which the annotator incrementally provides a new label to a few frames based on the current segmentation result. This will be tackled in our future work.

REFERENCES

[1] A. Hashimoto, N. Mori, T. Funatomi, Y. Yamakata, K. Kakusho, and M. Minoh, "Smart kitchen: A user centric cooking support system," in *Proc. 12th Int. Conf. Inf. Process. Manage. Uncertainty Knowl.-Based Syst.*, 2008, pp. 848–854.

[2] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using RGB-D," in *Proc. 14th ACM Conf. Ubiquitous Comput.*, 2012, pp. 208–211.

[3] S. Bansal, S. Khandelwal, S. Gupta, and D. Goyal, "Kitchen activity recognition based on scene context," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 3461–3465.

[4] S. Bianco, G. Ciocca, P. Napoletano, R. Schettini, R. Margherita, G. Marini, and G. Pantaleo, "Cooking action recognition with *i*VAT: An interactive video annotation tool," in *Proc. 17th Int. Conf. Image Anal. Process.*, 2013, pp. 631–641.

[5] S. Bianco, G. Ciocca, and P. Napoletano, "On the use of MKL for cooking action recognition," *Proc. SPIE*, vol. 9024, Mar. 2014, Art. no. 90240G.

[6] S. Michibata, K. Inoue, M. Yoshioka, and A. Hashimoto, "Cooking activity recognition in egocentric videos with a hand mask image branch in the multi-stream CNN," in *Proc. 12th Workshop Multimedia Cooking Eating Activities*, Jun. 2020, pp. 1–6.

[7] L. Wang, L. Cheng, and L. Wang, "Elastic sequence correlation for human action analysis," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1725–1738, Jun. 2011.

[8] K. Hara, K. Nakamura, and N. Babaguchi, "Temporal spotting of human actions from videos containing actor's unintentional motions," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2015, p. 6.

[9] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.

[10] B. Kruger, A. Vogele, T. Willig, A. Yao, R. Klein, and A. Weber, "Efficient unsupervised temporal segmentation of motion data," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 797–812, Apr. 2017.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[12] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[13] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1932–1939.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, p. 8.

[15] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 357–360.

[16] P. Agustí, V. J. Traver, and F. Pla, "Bag-of-words with aggregated temporal pair-wise word co-occurrence for human action recognition," *Pattern Recognit. Lett.*, vol. 49, no. 1, pp. 224–230, Nov. 2014.

[17] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5822–5831.

[18] J. Zhao and C. G. M. Snoek, "Dance with flow: Two-in-one stream action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9935–9944.

[19] A. Ramaswamy, K. Seemakurthy, J. Gubbi, and B. Purushothaman, "Spatio-temporal action detection and localization using a hierarchical LSTM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 764–765.

[20] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, Oct. 2005, pp. 65–72.

[21] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 299–318, Sep. 2008.

[22] K. Soomro and M. Shah, "Unsupervised action discovery and localization in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 696–705.

[23] G. Gong, X. Wang, Y. Mu, and Q. Tian, "Learning temporal co-attention models for unsupervised video action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9819–9828.

[24] M. Jain, J. van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek, "Tubelets: Unsupervised action proposals from spatiotemporal super-voxels," *Int. J. Comput. Vis.*, vol. 124, no. 3, pp. 287–311, Sep. 2017.

[25] P. Nguyen, B. Han, T. Liu, and G. Prasad, "Weakly supervised action localization by sparse temporal pooling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6752–6761.

[26] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, "3C-Net: Category count and center loss for weakly-supervised action localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8679–8687.

[27] M. Rashid, H. Kjellstrom, and Y. J. Lee, "Action graphs: Weakly-supervised action localization with graph convolution networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2020, pp. 615–624.
[28] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.
[29] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. 3rd Int. Workshop Similarity-Based Pattern Recognit.*, 2015, pp. 84–92.
[30] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 849–856.
[31] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, pp. 1–6, Dec. 2004.
[32] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. J. Comput. Commun.*, vol. 5, no. 1, pp. 27–34, 2011.

KAZUAKI NAKAMURA (Member, IEEE) received the B.S. degree in engineering and the M.S. and Ph.D. degrees in informatics from Kyoto University, in 2005, 2007, and 2011, respectively. Since 2012, he has been an Assistant Professor with the Graduate School of Engineering, Osaka University. His research interests include image processing, image recognition, and video analysis. He is a member of IEICE, IPSJ, and ITE.

NAOKO NITTA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in engineering from Osaka University, in 1998, 2000, and 2003, respectively. From 2002 to 2004, she was a Research Fellow with the Japan Society for the Promotion of Science. From 2003 to 2004, she was a Visiting Scholar with Columbia University. She is currently an Associate Professor with the Graduate School of Engineering, Osaka University. Her research interests include multimedia content and social media analysis.

NOBORU BABAGUCHI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Japan, in 1979, 1981, and 1984, respectively. He is currently a Professor and the Dean of the Graduate School of Engineering, Osaka University. He has recently been researching privacy protection methods for visual information and the security and fabrication of multimedia. He has published over 250 journal and conference papers and several textbooks. His research interests include image/video analysis, multimedia computing, and intelligent systems. He is a fellow of the IEICE, the Vice President of the ITE, and a member of the ACM, IPSJ, and JSAI. He also served as the General Co-Chair of MMM2008 and ACM Multimedia 2012, the Track Co-Chair of ICPR2012, the Area Chair of IEEE ICME2013, and the Honorary Co-Chair of ACM ICMR2018. He served as a Guest Editor for the IEEE Transactions on Information Forensics and Security, Special Issue on Intelligent Video Surveillance for Public Security & Personal Privacy.

KENSUKE FUJII joined the Komatsu Ltd., in 2016. He is currently assigned to Production Division, Manufacturing Engineering Development Center.

SATOKI MATSUMURA joined the Komatsu Ltd., in 2001. He is currently assigned to Production Division, Manufacturing Engineering Development Center. He is also responsible for the Team Leader of Construction Equipment System Team.

EIJI NABATA joined the Komatsu Ltd., in 1992. He is currently assigned to Production Division, Manufacturing Engineering Development Center. He is also responsible for the Group Manager of System Group.